Lecture Note on Python

WZ

(Dated: April 7, 2019)

This note includes everything covered in the class. It will be updated every week.

I. SYLLABUS

Objectives: In this class, you will learn the basics of Python. You will be able to write excutable codes on your own and read most python codes. Most importantly, you will learn how to treat and solve problem in the sense of computer science.

Instructor: WZ

Class Meetings:

regular meeting on Sunday 2:30-4:30pm;

to be scheduled for Monday 4:30-6:30pm.

Mar 31-Jun 10

20 lessons in 10 weeks, 2 hours for each lesson. In each lesson, the first half will focus on the lecture, and the second half for practice and tutoring.

This is an exercise-based class. The amount of knowledge you learned will be proportional to the amount of lines you wrote in Python. There will be assignment everyweek. We will have time to discuss or even do it in class. There will also be a project, to be started on week 5, finished on week 8, and revised in the last two weeks.

Textbooks:

For the lectures, we will follow these Slides.

For reference, this is a wikibook of python: Python for you and me.

Sample codes, assignments, and other material will be updated on GitHub.

The most powerful tool I have ever used is Google.

Prerequisite

We can start learning programming at any level, as long as you have stable access to a laptop for practicing.

Tentative content:

The following topics covers the basics in Python. As everyone has different needs when learning programming language. We will discuss how much of them will be covered and what to be added. Especially we can come up with some projects as preparation for the summer camp.

1. Introduction
2. gitHub, Functions, Booleans and Modules
3. Sequences, Iteration and String Formatting
4. Dictionaries, Sets, and Files
5. Exceptions, Testing, Comprehensions
6. Advanced Argument Passing, Lambda – functions as objects
7. Object Oriented Programming
8. More OO – Properties, Special methods
9. Iterators, Iterables, and Generators
10. Decorators, Context Managers, Regular Expressions, and Wrap Up

More topics to be determined

1. Graphics user interface(GUI).
2. Image/data processing
3. Internet communication, (TCP/IP)
4. Editors (emacs), window manager (tmux), typesetting format (latex, markdown)
5. Game engine (Unity)

CONTENTS

II. LECTURE NOTES

A. Lecture 1, Session 1, Mar 31, Introduction

Introduction to python
Course outline
Sample codes and projects
Environment setup (Installation, Editor, GitHub)
HelloWorld!

B. Lecture 2, Session 1, Apr 1, 2-4pm

start at assignment
finish on function

C. Lecture 3, Session 1-2, Apr 7, 2-4pm

review mac/linux,python,emacs review data type, function
start on function (return, parameter)
if, list, for
finish before variable scope, local v.s. global
exercise in class:FizzBuzz, exercise 2.2,2.3
home exercise: finish 2.2,2.3, do 2.4, think about 2.5

III. EXERCISE

For each exercise, please create a single `.py` file, e.g. `session1_1.py`. Please put all your files in one or several folders.

A. session 1

1. * create folder `firstFolder`, create file `first_file.txt`, write your name and date into that file; Copy this file to `second_file.txt`, and write "helloworld" into the second file.

   Hint: to copy a file in mac, use `cp [name of file1] [name of file2]`

2. * create helloworld.py; run it in python and shell respectively

3. * assign some value to a and b, calculate $c = a + b$ and print the result in the form "c=[value of c]"

4. * create a file to check the priority of `+-*/=`

5. * assign value of a,b,print them; switch value of them, and print the result

6. prepare a conversation between Jack and Rose, 5 sentence each.

   (This material will be used in several late programs: (a) print the conversation; (b) enunciate the conversation, using `say`.)

B. session 2

1. warmup exersice

2. (a) print sequence 1-100; (b) add sum of 1-100 and print the result

3. print sequence 100-1

4. (a) find average of three numbers (b) find maximum of three numbers

5. calculate prime numbers within 100

6. create random sequence of length 100

7. sort a sequence

8. calculate Pi

9. calculate square root of a number

10. game: 21 piont

C. ideas on big project

1. a conversation robot. Giving questions and responds according to answer
2. conversation application between two computer
3. a graphics program?
4. flappy bird
5. interactive math test. give random question and count the rate and time.
6. 2048

IV. BENCHMARK PROBLEM SET(TO BE UPDATED)

These problem ensure you to understand various concepts in Python. These are the source of assignment.

1. hello world!
2. summation of integers from 1 to 100
3. Fibonacci sequence
4. contact book
5. simple calculator
6. Tower of Hanoi
7. Tetris

V. COMMAND LIST

Limux/Mac terminal command

```
cd : enter a folder
ls : list of all the files in current folder
mkdir : create a new directly
rm : delete a file (be cautious!)
touch [file] : create file
less : view the file, press [q] to quit
cat : print the file
cp [file1] [file2] :copy file1 to file2
Tab : auto completement
man [cmd] : show the manual on how to use this command. for example
man cp
```

editor: emacs, prefix Ctrl-x or C-x

```
emacs + [filename] : open this file for editing
C-x C-s : save file
C-x C-c : exit
C-x C-f : find/open file
C-x C-w : write buffer to a new file
C-a : move cursor to the head of the line
C-e : move cursor to the end of the line
C-k : kill the line
C-y : yank/paste the lines has been killed right before
C-n : next line
C-p : previous line
```

how to run python in terminal

```
python : lauch python
python + [filename] : use python to excute this file
```

python command

```
print() : print string in ()
exit()  or Ctrl-d: exit python
```